Práctica #4: Uso de control de flujo

Introducción

El alumno creará cuatro programas en java que manejen control de flujo mediante las sentencias de salto If / else y Switch, así como de las sentencias de bucle for, while y do/while. Un ejemplo de cómo se utiliza el control de flujo en java es el siguiente:

```
//Programa que calcule los dias transcurridos
public class EjemploIF{
    public static void main(String args[]) {
       CalculaDias Dias = new CalculaDias();
       Dias.Inicializa(31,12);
       Dias.Calcula();
       Dias.Imprime();
       }
}
class CalculaDias{
       int diasT, dia, mes;
   public void Inicializa(int v1, int v2){
       dia = v1;
       mes = v2;    diasT = 0;
   }
   public void Calcula(){
       if(mes == 1)
          diasT = dia;
       if(mes == 2)
          diasT = 31 + dia;
       if(mes == 3)
          diasT = 59 + dia;
       if(mes == 4)
          diasT = 90 + dia;
       if(mes == 5)
          diasT = 120 + dia;
       if(mes == 6)
          diasT = 151 + dia;
       if(mes == 7)
          diasT = 181 + dia;
       if(mes == 8)
          diasT = 212 + dia;
       if(mes == 9)
          diasT = 243 + dia;
       if(mes == 10)
          diasT = 273 + dia;
       if(mes == 11)
          diasT = 304 + dia;
       if(mes == 12)
          diasT = 334 + dia;
   }
   public void Imprime(){
       System.out.println("Estamos a: " + dia + " / " + mes);
       System.out.println("Han Transcurrido: " + diasT + " dias");
   }
};
```

Fundamento Teórico

If / Else

```
if (expresión_booleana){
      bloque;
}
```

En el ciclo "if" el código se ejecutará siempre y cuando la *expresión_booleana* sea evaluada como verdadera, en caso de que sea evaluada como falsa, se podrá incluir un "else". Cuando se trate de un solo enunciado, se podrá hacer caso omiso de las llaves que delimitan al bloque. Además, se podrá hacer el uso de anidaciones, esto quiere decir que se puede poner un if, dentro del código de un if, tal como se muestra:

```
if (x == y)
      System.out.println("x es igual a y");
else
      if (x > y)
            System.out.println("x es mayor que y");
      else
            System.out.println("x es menor que y");
```

Switch

Permite que el programa ejecute diferentes bloques de instrucciones basado en una expresión que puede tener más de dos valores. Su sintaxis es la siguiente:

```
switch (variable){
      case caso_1:      bloque_1;
      break;
      case caso_2:      bloque_2;
      break;
      ...
      case caso_n:      bloque_n;
      break;
      [default: bloque_default;]
}
```

En donde la *variable* podrá ser de cualquier tipo y se compara contra cada caso:

- En caso de encontrar igualdad, la ejecución es transferida al *bloque* de enunciados que se encuentra a continuación del *caso* correspondiente.
- De no encontrar una igualdad, se ejecuta la opción default.
- Si no hay igualdad ni sentencia default, la ejecución pasa al primer enunciado después de la llave de cierre del switch.

El caso podrá estar constituido por una constante entera o por valores literales válidos, pero nunca variables o llamados a métodos. En caso de buscar coincidencia con caracteres, deberán estar encerrados en comillas simples. Es de suma importancia el agregar un enunciado break después del último enunciado de cada bloque, ya que de lo contrario todos los enunciados posteriores pertenecientes a los bloques restantes, se ejecutarán también.

Ciclo for

Este ciclo es una construcción que ejecuta un bloque de uno o más enunciados una determinada cantidad de veces. En Java, tiene la siguiente estructura:

```
for(inicial; expresión_booleana; incremento){
      bloque;
}
```

```
for (int count = 1; count <= 20; count++)
      System.out.println(count);
```

En donde:

- inicial será por lo general una expresión de asignación que ponga una variable a un valor determinado de inicio y dicha variable será típicamente la que controle el ciclo.
- expresión_booleana será la expresión a evaluar; el ciclo se ejecutará mientras sea verdadera y terminará cuando sea falsa.
- incremento será la expresión que determine el incremento o decremento de la variable controladora.
- bloque será el o los enunciados a realizar en el ciclo.

La flexibilidad del ciclo for permite que la variable controladora no necesariamente debe ser inicializada dentro del ciclo, además, el incremento o decremento puede ser hecho en cualquier otra parte del ciclo:

```
int count = 1;
for ( ; count < 1000; count++)
      System.out.println(count++);
```

Al igual que una instrucción condicional, un ciclo for puede ser ejecutado dentro de otro, logrando así una construcción más compleja y efectiva.

Objetivos de Aprendizaje

El propósito de esta práctica es que el estudiante se familiarice con el entorno de la programación en Java, mediante el uso de comandos básicos al utilizar el control de flujo al utilizar métodos con sentencias de salto y sentencias de bucle.

Descripción de la práctica

1. Crear dentro de la carpeta "**Programación**" otra carpeta con el nombre de "**Practica4**".
2. Abrir un nuevo documento en el block de notas y realizar el programa llamado **"EjemploIF"**, guardar el archivo como **.java** en la carpeta de "**Practica4**". Compilar el archivo mediante la línea de comandos, en caso que el compilador marque un error, corregir el código fuente, y compilar nuevamente. Si no hay error en el código fuente, ejecutar el programa.
3. Abrir un nuevo documento en el block de notas y realizar el programa llamado **"EjemploSwitch"**, lo que tienen que realizar es modificar el programa de EjemploIF y adaptarlo con el uso de Switch, guardar el archivo como **.java** en la carpeta de "**Practica4**". Compilar el archivo mediante la línea de comandos, en caso que el compilador marque un error, corregir el código fuente, y compilar nuevamente. Si no hay error en el código fuente, ejecutar el programa.
4. Abrir un nuevo documento en el block de notas y realizar el programa llamado **"EjemploFor"**, lo que tienen que realizar es crear un arreglo de 10 posiciones y llenarlo mediante el uso del bucle for, guardar el archivo como **.java** en la carpeta de "**Practica4**". Compilar el archivo mediante la línea de comandos, en caso que el compilador marque un error, corregir el código fuente, y compilar nuevamente. Si no hay error en el código fuente, ejecutar el programa.
5. Abrir un nuevo documento en el block de notas y realizar el programa llamado **"CatalogoRopa"**, lo que tienen que realizar es modificar el programa realizado en la práctica 3 llamado "Catalogo" y adaptarlo utilizando las sentencias de ciclo y de bucle vistas en la práctica. Guardar el archivo como **.java** en la carpeta de "**Practica4**".

Compilar el archivo mediante la línea de comandos, en caso que el compilador marque un error, corregir el código fuente, y compilar nuevamente. Si no hay error en el código fuente, ejecutar el programa.

6. Realizar un reporte con los resultados obtenidos. El reporte debe contener Número de práctica, nombre de la experiencia educativa, nombre del alumno, matrícula, fecha de elaboración, desarrollo de la práctica (Código Fuente), y resultados.

Tarea 4

Retomar el código fuente de "**Elevador**", utilizar un if/else el cual envíe instrucciones para abrir y cerrar puertas, además de un switch para indicar a que piso se requiere ir, también el programa debe indicar si el elevador sube o baja. Guardar el archivo como **Elevador2.java** en la carpeta de "**Practica4**". Compilar el archivo mediante la línea de comandos, en caso que el compilador marque un error, corregir el código fuente, y compilar nuevamente. Si no hay error en el código fuente, ejecutar el programa.

Materiales a emplear en la práctica

- Computadora con el JDK instalado.
- Block de notas
- Línea de comandos o terminal.

Bibliografía

Froute Quintas, A. (2008). *JAVA 2: Manual de usuario y tutorial* (5th ed.). México: Alfaomega, ISBN 978-970-15-1430-6.